Open vSwitch and Cloud

Portions of this PPT draw from PPT authored by Professor Dijiang Huang at Arizona State University

ASU ARIZONA STATE UNIVERSITY

Outline

- Concept of SDN
- OpenFlow – a SDN Implementation
- **Open vSwitch**

Open vSwitch

- Open vSwitch is a multilayer software switch that resides within the hypervisor and provides connectivity between the virtual machines and the physical interfaces.
- It provides interfaces for manipulating the forwarding state and managing configuration state at run-time.
- The 3 interfaces are :
  - **Configuration interface**
  - **Forwarding path interface**
  - **connectivity management interface**.

Open vSwitch

- Configuration interface:
  - A remote process can read and write configuration state (as key/value pairs), and set up triggers to receive asynchronous events about configuration state changes.
  - Bond interfaces for improved performance and availability.
  - Provides bindings between network ports and the larger virtual environment.

Open vSwitch

- Forwarding path interface:
  - Allows an external process to write the forwarding table directly.
  - The lookup can decide to forward the packet out of one or more ports, to drop the packet, or to en/decapsulate the packet.
  - Implements a superset of the OpenFlow protocol.

Open vSwitch

- Management interface:
  - Virtualization layer can manipulate its topological configuration. Ex: Creating switches.
  - Managing VIF and PIF connectivity
- In its simplest deployment Open vSwitch is a traditional physical switch within the virtualization layer.
- Enables distribution of the switch functions across multiple servers decoupling the logical network topology from the physical one.

Open vSwitch

- Centralized Management
- Virtual Private Networks
- Mobility between IP subnets

Open vSwitch

- Centralized Management
  - The interfaces provided by Open vSwitch can be used to create a single logical switch image across multiple Open vSwitches running on separate physical servers.
  - Therefore, as VMs join, leave, and migrate, it is the responsibility of this management process to ensure any configuration state remains coupled to the logical entities.
  - It is possible to query and configure a collection of virtual switches as if they were a single switch.

Open vSwitch

- Virtual Private Networks
  - Collection of VMs can be connected to each other over a private, virtual network implemented on top of a shared physical network infrastructure.
  - VMs sharing a private network spread across multiple hosts/physical switches, requires virtualization networking layer to support dynamic overlay creation.
  - Open vSwitch uses tunnels (GRE) to encapsulate an Ethernet frame inside an IP datagram to be routed.

Open vSwitch

- Virtual Private Networks
  - A global management process can select the best way to forward packets from one VM to another modifying flow tables accordingly in Open vSwitches:
    - Virtual private network on the same Open vSwitch
    - VLANs (same subnet)
    - GRE tunnels (multiple subnets)

Open vSwitch

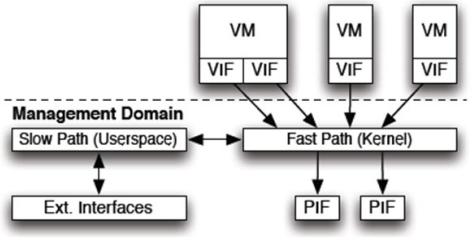
- Mobility between IP subnets
  - Well known limitation of virtualization platform is that migration must happen within an IP subnet.
  - But, migration between subnets is desirable as single L2 domains have scalability limits.
  - A model similar to Mobile IP can be used, where a base Open vSwitch can receive all packets for a VM and forwarding the packet to its true location using tunneling.

Open vSwitch

- Open vSwitch implementation consists of two components:
  - kernel-resident "fast path"
  - userspace "slow path"

Open vSwitch

IMPLEMENTATION



Open vSwitch

IMPLEMENTATION

- Fast path implements forwarding engine which is responsible for per-packet lookup, modification and forwarding.
- Majority of functions is implemented within slow path running in the VM management domain (Dom0).
  - Implements forwarding logic
  - MAC learning
  - Load balancing
  - Remote visibility – OpenFlow, NetFlow etc.

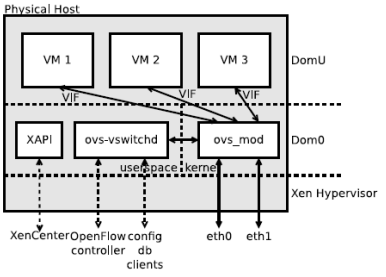
Open vSwitch

IMPLEMENTATION

- Fast path, being speed critical portion of the system has 3000 lines of code within the kernel and is system-specific.
- Open vSwitch emulates Linux bridging code and can be used as a replacement for virtual switches used by XenServer.

Open vSwitch

- Open vSwitch integration with XenServer

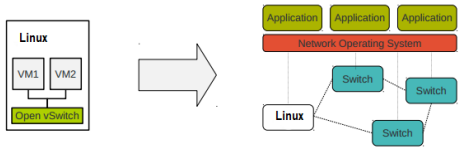


Open vSwitch

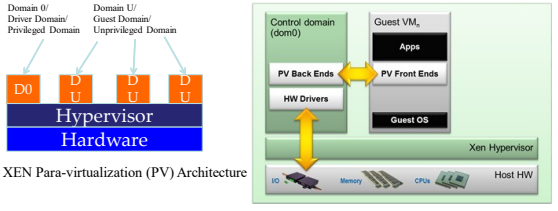
- Open vSwitch integration with XenServer
  - Open vSwitch works seamlessly with XenServer, will ship with Open vSwitch as the default.
  - XAPI is responsible for managing all aspects of a XenServer.
  - Notifies Open vSwitch of events related to network configuration.
  - Notifies Open vSwitch when bridges should be created and interfaces should be attached to them.
  - Open vSwitch stores this information in its configuration database, which notifies any remote listeners, such as a central controller.

Why Open vSwitch

- Open vSwitch enables Linux to become part of a SDN architecture.



Software Switch Networking I/O Architecture

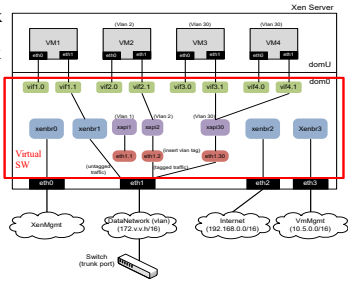


XEN Para-virtualization (PV) Architecture

19

Xen & Virtual Software Networking

- The old version of Citrix XenServer (before v5.6 FP1) using simple Linux Bridge.
- Many hypervisor based virtualization also apply Linux Bridge model, such as KVM, libvirt.
- All of bridging work are done by 'brctl'.
- Provide simple L2 switching functions.



20

Open vSwitch's Features

- Visibility:
  - NetFlow, sFlow, Mirroring (SPAN/RSPAN/ERSPAN)
- Control:
  - Centralized control through OpenFlow
    - o Missed flows go to central controller
  - Fine-grained ACL and QoS (Quality of Service) policies
    - o L2-L4 matching and actions to forward, drop, modify, and queue
- Forwarding:
  - LACP (Link Aggregate Control Protocol)
  - Port bonding
  - Standard 802.1Q VLAN model with trunk and access ports
  - GRE, GRE over IPSEC, Ethernet-over-GRE and CAPWAP tunneling
- Compatibility layer for Linux bridging code
- High-performance forwarding using a Linux kernel module

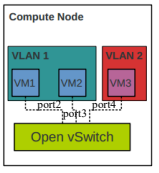
22

Feature – security/L2 segregation

- VLAN isolation enforces VLAN membership of a VM without the knowledge of the guest itself.

```
# ovs-vsctl add-port ovsbr port2 tag=10
```

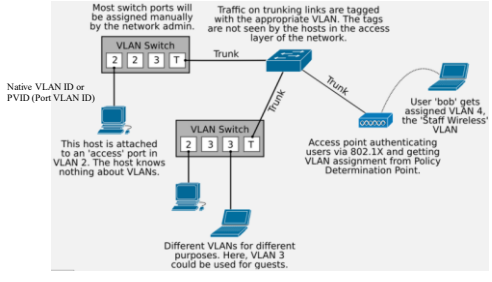
Any limit for VLAN ID?



23

What is VLAN?

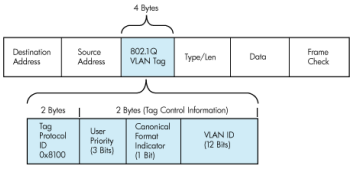
- A Virtual LAN (VLAN) is the ability to segregate a switch into separate broadcast-domains.



24

IEEE 802.1Q

- The standard defines a system of VLAN tagging for Ethernet Frames.



VLAN ID limit: 2^12

25

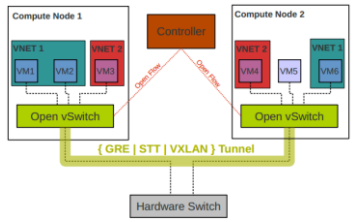
Benefit of VLANs

- VLANs give us three major benefits:
  - traffic control by prioritizing traffic in particular VLANs or reducing broadcast traffic by making the broadcast domains smaller
  - security, by controlling traffic between different VLANs (subnets),and
  - flexibility in network design without extra equipment.
- What is difference between Subnet and VLAN?

26

Feature – Tunneling

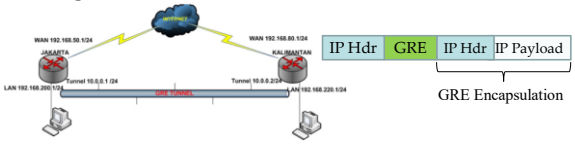
- Tunneling provides isolation and reduces dependencies on the physical network.



27

What is GRE Tunnel?

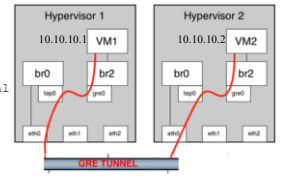
- A Tunneling protocol that was developed by Cisco.
- Generic routing encapsulation (GRE) can encapsulate a variety of protocol packet types inside IP tunnels.
- This creates a virtual point-to-point link to Cisco routers at remote points over an IP network.
- GRE tunneling is a layer 3 technology and as such requires a layer 3 device such as a router or layer 3 capable switch.



28

Create GRE Tunnel with OVS

```
# Create an Isolated Bridge
> ovs-vsctl add-br br2
# Create the GRE Tunnel Endpoint
> ovs-vsctl add-port br0 tep0 \
  -- set interface tep0 type=internal
# Assign it with an IP address
> ifconfig tep0 192.168.100.10/24
# Establishing the GRE Tunnel
> ovs-vsctl add-port br2 gre0 \
  -- set interface gre0 type=gre \
  Options:remote_ip=192.168.200.10/24
# Repeat these commands on the other hypervisor
```



29

Feature – Visibility

- Support industry standard technology to monitor the use of a network.
  - sFlow
  - NetFlow
  - Port Mirroring
    - o SPAN
    - o RSPAN
    - o ERSPAN

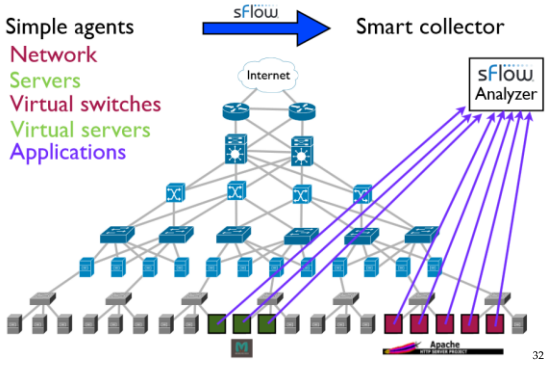
30

What is sFlow/NetFlow?

- **sFlow** is a technology for monitoring network, wireless and host devices.
- Flow samples: based on a defined sampling rate, an average of 1 out of n packets/operations is randomly sampled. This type of sampling does not provide a 100% accurate result, but it does provide a result with quantifiable accuracy.
- Counter samples: A polling interval defines how often the network device sends interface counters.
- sFlow datagrams: The sampled data is sent as a UDP packet to the specified host and port (6343).

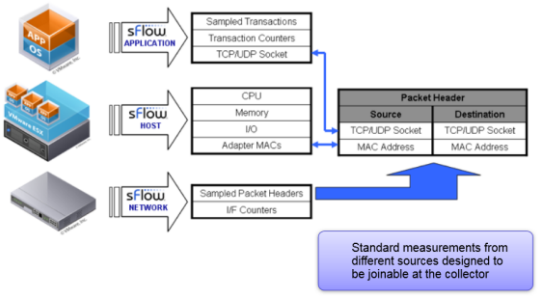
31

sFlow Architecture



32

Cross-layer correlation: Application, Host and Network



33

Forwarding Components

- Forwarding Components
  - ovs-vswitchd (control plane, slow path)
    - A daemon that implements the switch, along with a companion Linux kernel module for flow-based switching.
    - Forwarding logic (learning, mirroring, VLANs, and bonding)
    - Remote configuration and visibility
  - openvswitch_mod.ko (data plane, fast path)
    - Packet lookup, modification, and forwarding
    - Tunnel encapsulation/decapsulation

34

Other Modules and Tools

- ***ovsdb-server***: a lightweight database server that ovs-vswitchd queries to obtain its configuration.
- ***ovs-brcompatd***: a daemon that allows ovs-vswitchd to act as a drop-in replacement for the Linux bridge in many environments, along with a companion Linux kernel module to intercept bridge ioctls.
- ***ovs-dpctl***: a tool for configuring the switch kernel module.
- ***ovs-vsctl***: a utility for querying and updating the configuration of ovs-vswitchd.

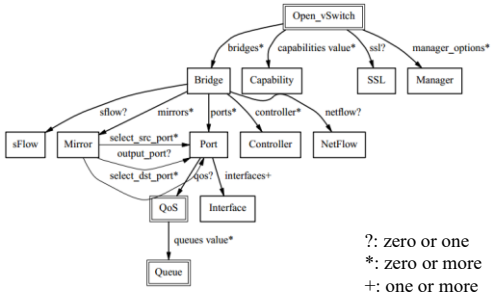
35

Other Modules and Tools

- ***ovs-appctl***: a utility that sends commands to running Open vSwitch daemons.
- ***ovs-ofctl***: a utility for querying and controlling OpenFlow switches and controllers.
- *Ovsdbmonitor*: a GUI tool for remotely viewing OVS databases and OpenFlow flow tables.
- *ovs-controller*: a simple OpenFlow controller.
- *ovs-pki*: a utility for creating and managing the public-key infrastructure for OpenFlow switches.

36

OVSDB Table Relationships



?: zero or one
*: zero or more
+: one or more

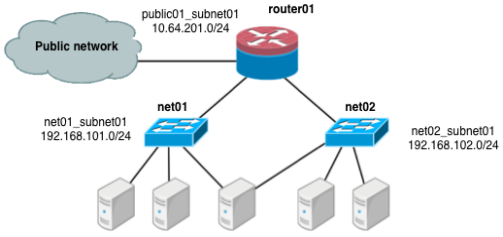
37

OVS Example

```
# Show the current bridges and the attached ports,interfaces
> ovs-vsctl show

# Show OpenFlow information
> ovs-ofctl show br-int

# print all flow entries
> ovs-ofctl dump-flows br-int

# add flows
> ovs-ofctl add-flow br1 nw_src=192.168.1.2,nw_dst=192.168.2.2, \
   idle_timeout=0,icmp,action=mod_nw_src:172.16.206.2, \
   mod_nw_dst:172.16.121.2,output:0
> ovs-ofctl add-flow br1 nw_src=192.168.1.2,nw_dst=192.168.2.3, \
   idle_timeout=0,icmp,action=mod_nw_src:172.16.206.2, \
   mod_nw_dst:172.16.121.3,output:0
```


OVS in OpenStack

- Scenario: one tenant, two networks, one router



References

1. Scott Shenker, Software Defined Networking, http://inst.eecs.berkeley.edu/~ee122/
2. OpenFlow official website at http://www.openflow.org
3. NOX controller at http://www.noxrepo.org/
4. J. Pettit, J. Gross, B. Pfaff, M. Casado, S. Crosby, "Virtual Switching in an Era of Advanced Edges," 2nd Workshop on Data Center – Converged and Virtual Ethernet Switching (DC-CAVES), ITC 22, Sep. 6, 2010.
5. B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, S. Shenker, "Extending Networking into the Virtualization Layer," HotNets-VIII, Oct. 22-23, 2009.
6. S. Zhou, "Virtual Networking," ACM SIGOPS Operating Systems Review, 2010.
7. N. Mckeown et al., "OpenFlow: Enabling Innovation in Campus Networks," white papers. The OpenFlow Switch Consortium, March 2008, http://www.openflowswitch.org//documents/openflow-wp-latest.pdf
8. Y. Luo et al., "Accelerating OpenFlow Switching with Network Processors," Proc. ACM/IEEE Symp. Architectures for Networking and Communications Systems (ANCS 09), ACM Press, 2009.
9. J. Naous et al., "Implementing an OpenFlow Switch on the NetFPGA platform," Proc. ACM/IEEE Symp. Architectures for Networking and Communications Systems (ANCS 08), ACM Press, 2008.
10. H. Chen et al., "A Survey on the Application of FPGAs for Network Infrastructure Security," IEEE Communications Surveys & Tutorials, June 2010.
11. F. Azmandian et al., "Virtual Machine Monitor-Based Lightweight Intrusion Detection," ACM SIGOPS Operating Systems Review, July 2011.
12. H. Bos and K. Huang, "A network intrusion detection system on ixp1200 network processors with support for large rule sets," in Technical Report 2004-02. Leiden University, 2004.

Questions ?

ASU ARIZONA STATE UNIVERSITY